UNITED STATES PATENT APPLICATION FOR:

SYSTEM AND METHOD FOR DOMAIN CONFIGURATION

Inventors:

David Felts
Thomas Nugent
Alexander Toussaint

SYSTEM AND METHOD FOR DOMAIN CONFIGURATION

Inventors:

David Felts
Thomas Nugent
Alexander Toussaint

COPYRIGHT NOTICE

[0001]   A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

CLAIM OF PRIORITY

[0002]   This application claims priority from the following application, which is hereby incorporated by reference in its entirety:

[0003]   SYSTEM AND METHOD FOR DOMAIN CONFIGURATION, U.S. Application No. 60/450,126, Inventors: David Felts et al., filed on February 25, 2003. (Attorney's Docket No. BEAS-1383US0)

FIELD OF THE DISCLOSURE

[0004]   The present disclosure relates generally to interactive graphical tools for configuring network resources and, in particular, application/web server configuration tools.

BACKGROUND

[0005]   As the use of network domains for management, load-balancing and fail-over increases, so to does the need for easy-to-use domain management tools. Some domain management tools have rigid user interfaces that only accommodate a predefined set of domain configurations. What is needed is an interactive domain tool that is easy to extend.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006]   **Figure 1** is an exemplary illustration of a configuration wizard/user interface user interface in accordance to an embodiment.

[0007]   **Figure 2** is an exemplary illustration of a domain in an embodiment.

[0008]   **Figure 3** is an exemplary script for creating a new domain from a configuration template in an embodiment.

[0009]   **Figure 4** is an exemplary script for creating a new domain and updating it using an extension template

DETAILED DESCRIPTION

[0010]   Aspects of the invention are illustrated by way of example and not by way of limitation in the figures of the accompanying drawings in which like references indicate similar elements. It should be noted that references to "an" or "one" embodiment in this disclosure are not necessarily to the same embodiment, and such references mean at least one.

[0011]   A domain can be defined in which one or more enterprise applications can execute. A domain is the basic administration unit and can include web/application server(s), portal(s), business-to-business integration products, interactive development tool(s), and enterprise applications. In one embodiment, domain creation is based on one or more templates. Templates can contain domain, application and service configuration parameters along with supporting files.

[0012]   **Figure 1** is an exemplary illustration of a configuration wizard/user interface user interface in accordance to an embodiment. The configuration wizard/user interface can be used to install, configure, monitor, and/or manage one or more domains. In one embodiment, the wizard/user interface can be a Java™ application. The Configuration Wizard/user interface can guide a user through the process of creating or extending a domain for a target environment. This process is based on predefined template(s) that contain the attributes and files required for building or extending a particular domain. In one embodiment, a template can be a Java™ Archive (JAR) file that contains the files and scripts required to create or update a domain.

[0013]   In one embodiment, the types of templates used by the configuration wizard/user interface can include configuration and extension templates.   A

configuration template can define the full set of resources within a domain, including infrastructure components, applications, services, security options, and general environment and operating system options. This type of template is used to create a domain. An extension template can define applications and services, such as JDBC (Java™ Database Connectivity) or JMS (Java™ Message Service) components, and startup/shutdown classes, that can be used to extend an existing domain. This type of template is used to update a domain. The applications and services stored in the selected extension templates can be imported into the domain.

[0014]   The Configuration Wizard/user interface provides two options for creating domains: express and custom configuration. Express configuration allows a user to create a domain quickly, using default settings from a selected configuration template. Custom configuration allows a user to modify configuration information. To customize a domain, a user can specify any of the following components and parameters: Infrastructure components, including managed servers, clusters, and physical host machines; database and messaging services—Java Database Connectivity (JDBC) and Java Message Service (JMS); Targets (servers and clusters); security parameters; and general environment and operating system parameters.

[0015]   In various embodiments, the Configuration Wizard/user interface can create a domain that includes a configuration file, `config.xml`, that can describe the infrastructure and basic network parameters of all server instances, and configuration of basic security features that allow for the initial booting of the domain. The domain may also include the following:

- Server startup scripts (by way of a non-limiting example, `startWebLogic.cmd` or `startWebLogic.sh`) that are populated with the values that a user enter using the Configuration Wizard/user interface;

- A directory containing the applications provided by the template;

- A directory containing database-specific information used to create and initialize a database; and

- Other files and directories to help a user get started (by way of a non-limiting example, `setEnv.cmd` or `setEnv.sh`).

[0016]   In some cases, it may be desirable or necessary to add a predefined application, a component product, or a set of services, such as JDBC or JMS, to an

existing domain. By way of a non-limiting example, if a user has an existing Web/app server application running in a domain but needs to develop an Integrated Development Environment (IDE) application to run in the same domain, a user can extend the functionality of the Web/app server domain by adding the IDE.

[0017] In one embodiment, to extend a domain using the Configuration Wizard/user interface, a user can select the directory of the domain that the user wants to extend and specify that the extension template include additional applications. A user then has the option of configuring the database (e.g., JDBC) and messaging (e.g., JMS) services, and targeting servers or clusters to which a user want the applications or services deployed. The Configuration Wizard/user interface can then update the config.xml file and all other generated components in the configuration directory, as required.

[0018] **Figure 2** is an exemplary illustration of a domain in an embodiment. A domain is the basic administration unit for Web/app server. It consists of one or more Web/app server instances **200**, and logically related resources and services that are managed, collectively, as a unit. The basic domain infrastructure consists of one Administration Server **202** and optional managed servers **200** and clusters **204**. In one embodiment, a domain can include one web/app server instance that is configured as an Administration Server. The Administration Server provides a central point for managing the domain and providing access to the Web/app server administration tools **205**. These tools include, but are not limited to, the following: a graphical administration console and a server node manager that enables a user to start, shutdown, restart and monitor servers.

[0019] Managed Servers can host application components and resources, which are also deployed and managed as part of the domain. In a domain with only a single Web/app server instance, the server can function as both the Administration Server and Managed Server. A domain may also include Web/app server clusters, which are groups of Web/app server instances that work together to provide scalability and high availability for applications. Clusters can improve performance and provide fail-over should a server instance become unavailable. The servers within a cluster can run on the same machine, or they can reside on different machines. To a client, a cluster appears as a single Web/app server instance.

[0020] In addition to infrastructure components, a domain can define the basic network configuration for the server instances it contains. Specifically, a domain can define application deployments, supported application services (such as database and messaging services), security options, and physical host machines.

[0021]   A user may find it useful to configure multiple domains based on specific criteria, such as system administrator responsibilities, the logical classification of applications, the geographical locations of servers, or size. The following table outlines the most common domain configurations.

FEATURE	DESCRIPTION
Domain with Managed Servers	In typical production environments, several Managed Servers are deployed to host Web applications, and an Administration Server is deployed to perform management operations.
Domain with Managed Servers and Clusters	In production environments that require increased performance, Cluster throughput, or availability for a Web application, several Managed Servers might be grouped in a cluster. In such a case, the domain consists of one or more clusters with the applications they host, additional Managed Servers, if necessary, and an Administration Server to perform management operations.
Stand-alone Server Domain	In development or test environments, a single application and server might be deployed independently from other Managed Servers. In such a case, a user can deploy a domain consisting of a single Administration Server that also hosts the Web applications a user want to test or develop. The Web/app server Examples Domain template, described in the Template Reference, is an example of a stand-alone server domain.

Table 1: Common Domain Configurations

[0022]   In one embodiment, the Configuration wizard/user interface can be invoked in graphical mode or silent mode. To run it in silent mode, a user can first create a script that can define the configuration settings for the domain. A script can specify operations to be performed by the Configuration Wizard/user interface when the script is executed. The following sections describe the operations that can be defined in the script to create a new domain.First, a pre-existing configuration template can be selected to use as the basis of the domain a user will create and configure. Then' the information in the template can be customized to create new configuration objects, edit existing values for configuration object attributes, or delete configuration objects. In one embodiment, a user can obtain information about a user existing configuration by viewing the `config.xml` and `security.xml` files in a user template or domain. In one embodiment and by way of a non-limiting example, the following table can define the configuration objects that can be created and edited.

CONFIGURATION OBJECT TYPE	WHAT IS DEFINED?
Server	Administration or Managed Server.
Cluster	Cluster.
Machine	Windows machine.
UnixMachine	UNIX machine.
JDBCConnectionPool	JDBC Connection Pool.
JDBCMultiPool	JDBC MultiPool.
JDBCDataSource	JDBC data source.
JDBCTxDataSource	JDBC data source that supports global transactions.
JMSConnectionFactory	JMS connection factory.
JMSDestinationKey	JMS destination key.
JMSServer	JMS Server.
JMSFileStore	JMS file store.
JMSJDBCStore	JMS JDBC store.
JMSTemplate	JMS Template.
JMSTopic	JMS topic.
JMSQueue	JMS queue.
JMSDistributedTopic	JMS distributed topic.
JMSDistributedQueue	JMS distributed queue.
JMSDistributedTopicMember	JMS distributed topic members.
JMSDistributedQueueMember	JMS distributed queue members.
User	User.
Group	Group.

Table 2: Exemplary Configuration Object Types in an Embodiment

[0023]   A configuration is specified in a template file as follows:

```
create object_type "name" as variable;
```

where *object_type* is the configuration object type (see **Table 2**), *name* is how the object will be referred to, and *variable* is a user-defined variable used to specify the configuration object.

[0024]   In the following example, a Server object is created and assigned to the variable s2, and the Name attribute is set to server_2:

```
create Server "server_2" as s2;
```

[0025]   In the following example, a User object is created and assigned to the variable u1, and the Name attribute is set to user1:

```
create User "user1" as u1;
```

[0026] In one embodiment and by default, the user is assigned to the group "Administrators."

[0027] Before a user can edit an existing configuration object, the configuration wizard/user interface can find the object in the template and assign it to a variable. To assign a configuration object to a variable, use the find operation, as follows:

```
find object_type "name" as variable;
```

where *object_type* is a configuration object, *name* is the value of the configuration object Name attribute, and *variable* is a user-defined variable used to specify the configuration object.

[0028] In the following example, a Server object named "myserver" is assigned to the variable s1:

```
find Server "myserver" as s1;
```

[0029] To set a configuration attribute, specify the variable and attribute name for it, as follows:

```
set variable.attribute "value";
```

where *variable* is a user-defined variable used to specify the configuration object, *attribute* is a configuration object attribute value, and *value* is a value that a user want to assign to the specified attribute of the configuration object. The value can be enclosed in quotation marks.

[0030] **Figure 3** is an exemplary script for creating a new domain from a configuration template in an embodiment.

[0031] **Figure 4** is an exemplary script for creating a new domain and updating it using an extension template

[0032] **Table 3** summarizes the procedure for creating a domain using the custom configuration option in an embodiment Configuration wizard/user interface.

TASK	DESCRIPTION
Creating or Extending a Configuration	Choose whether to create a new WebLogic domain configuration or add to an existing domain configuration. To create a new domain, choose Create a new WebLogic configuration.
Selecting a Configuration Template	Choose the configuration template with which a user want to create and configure a user domain.
Choosing Express or Custom Configuration	Select the Custom option to create a domain with non-default template settings that meet the needs of a user domain.
Designating an Administration Server	Designate a server as the Administration Server. This window is displayed automatically only if the selected template includes multiple servers that are not assigned to clusters.
Configuring the Administration Server	Define parameters for the Administration Server, a designated server from which the domain is managed.
Configuring Managed Servers, Clusters, and Machines When Creating a New Domain	Optionally, define parameters for Managed Servers, clusters, and host machines.
Configuring JDBC When Creating a New Domain	Optionally, define parameters for Java Database Connectivity (JDBC).
Configuring JMS When Creating a New Domain	Optionally, define parameters for the Java Message Service (JMS).
Targeting Servers and Clusters When Creating a New Domain	Optionally, target the servers and clusters to which a user want to deploy application components (such as Web applications and EJB modules), and application services (such as JDBC or JMS components, and startup/shutdown classes.)
Configuring Security When Creating a New Domain	Specify an administrative username and password. Optionally, a user can also configure additional security features by defining users and groups and assigning them to global security roles.
Configuring Windows Options	Optionally, define domain-specific parameters for the Windows operating system.
Specifying the WebLogic Configuration Environment	Select the mode in which to launch a user WebLogic domain configuration (development mode or production mode) and the Java Software Development Kit (SDK) that is enabled for the selected Startup mode.

TASK	DESCRIPTION
Creating the WebLogic Configuration	Review the parameters defined for a user domain configuration, specify its name and pathname, and initiate its creation.

Table 3: Exemplary Tasks for Creating a Custom Domain in an Embodiment

[0033]   In various embodiments, the configuration wizard/user interface can prompt a user to choose whether a user want to create a new domain configuration or extend an existing one by adding applications and services. To create a new domain, a user begins by selecting a configuration template as the basis for a user domain and then modifies settings as needed. To add applications and services to an existing domain configuration, a user begins by selecting the extension template to modify.

[0034]   The configuration wizard/user interface can prompt a user to define configuration information for the Administration Server.   A user can review the values displayed in the window and modify them as necessary, using the guidelines provided in the following table.   Exemplary values in one embodiment are provided in **Table 4**.

FIELD	DESCRIPTION
Name	A valid server name: a string of characters that can include spaces. Each server instance in a user environment can have a unique name, regardless of the domain or cluster in which it resides, or whether it is an Administration Server or a Managed Server. In addition, the name of the Administration Server can be unique among all configuration component names within the domain.
Listen Address (Optional)	From a drop-down list, a user can select a value for the Listen Address. Valid values for the listen address are as follows: • All Local Addresses (default) • IP address of the computer that hosts the server • DNS name that resolves to the host • localhost (valid only for requests that are issued from the computer on which the server is running)
Listen Port (Optional)	A valid value for the listen port to be used for regular, non-secure requests (via protocols such as HTTP and T3). The default value is 7001. If a user leave this field blank, the default value is used.
SSL Listen Port (Optional)	SSL Listen Port Enter a valid value to be used for secure requests (via protocols such as HTTPS and T3S). The default value is 7002. If a user leave this field blank, the default value is used.
SSL Enabled (Optional)	Select the check box if a user want to enable the SSL Listen Port. By default, the SSL is disabled for all new servers.

Table 4: Exemplary Admin Server Configuration Parameters in an Embodiment

[0035] In one embodiment, the minimum requirement for a user domain is a single Administration Server on a single machine. In addition, however, a user has the option of configuring other resources to be managed by the Administration Server and distributing them across multiple machines. A user can: add, change, or delete managed servers; add, change, or delete clusters; group managed servers into clusters, or change current groupings; and assign servers to machines, or change current assignments. The configuration wizard/user interface can prompt a user to specify whether a user want to distribute configuration across Managed Servers, clusters, and physical machines.

[0036] JDBC enables Java™ programmers to interact with common database management systems (DBMS), such as Oracle, Microsoft SQL Server, and Sybase. The Configuration Wizard/user interface gives a user the option of configuring a database service for a user domain by defining several WebLogic JDBC components: Connection pools—Ready-to-use groups of connections to a user DBMS; MultiPools—Groups of connection pools; and Data sources—Interfaces between applications and connection pools. Because Web/app server applications usually require some database access, templates created for use with the Configuration Wizard/user interface can contain JDBC configuration information. When using the Configuration Wizard/user interface, a user can accept these settings, modify them, or add JDBC configuration information.

[0037] JMS gives a user access to enterprise messaging systems that enable applications to communicate with one another. The Configuration Wizard/user interface gives a user the option of setting up JMS messaging services for a domain. To set up a messaging service, a user can define the following components:

- Connection factory—Encapsulated connection configuration information that enables JMS applications to create a connection.
- File and JDBC stores—Disk-based file stores and JDBC-accessible database stores, respectively; used to store persistent messages.
- Destination Keys and JMS Templates—Keys to define the sort order for messages that arrive at a destination and templates to define multiple destinations with similar attribute settings.
- JMS Servers—Servers that manage connections and message requests on behalf of clients.
- Destinations—Queues (in point-to-point models) or topics (in publish/subscribe models) that serve as destinations and distributed destinations for a JMS server.

[0038] Because JMS is often required by Web/app server applications, the

templates created for use with the Configuration Wizard/user interface can contain JMS configuration information. A user can accept these settings when using the Configuration Wizard/user interface.

[0039] In one embodiment, the configuration wizard/user interface can prompt a user to decide whether a user want to define additional infrastructure components and distribute the domain across those components.

- If a user selects No, the wizard/user interface uses the configuration settings based on a configuration source that a user selected earlier.

- If a user selects Yes, the user is prompted in subsequent windows to define the configuration a user want, as described in the **Table 5**.

TASK	DESCRIPTION
Configuring Managed Servers	Optionally, add managed servers or change the configuration of existing managed servers in a user domain.
Grouping Managed Servers into Clusters	A cluster is a group of Web/app server instances that work together to provide scalability and high-availability for applications.
Mapping Web/app server Instances to Host Machines	In a domain, a machine is the computer hardware that hosts one or more Web/app server instances.

Table 5: Exemplary Tasks for Configuring Servers and Clusters in an Embodiment

[0040] To configure clusters, a user can review a current list of cluster configurations and add or modify entries as required using the guidelines provided in **Table 6**.

FIELD	DESCRIPTION
Name	Valid cluster name: a string of characters that can include spaces. The name of the cluster can be unique among all configuration component names within the domain.
Multicast Address (Optional)	Multicast address for the cluster. This address is used by cluster members to communicate with each other. The default value is 237.0.0.1. Valid multicast addresses are any valid IP address of the form nnn.xx.xx.xxxx, where nnn is 237, 238, or 239.
Multicast Port	Multicast port for the cluster.The multicast port is used by cluster

FIELD	DESCRIPTION
(Optional)	members to communicate with each other. The default value is 7777.
Cluster Address (Optional)	Cluster address that identifies the Managed Servers in the cluster. A cluster address can be one of the following: ○ Comma-separated list of IP addresses or DNS names and ports, by way of a non-limiting example: dns_name:port, dns_name:port ○ DNS name that maps to multiple IP addresses ○ localhost, DNS name, or IP address if the Listen Address of all Managed Servers is listening to the same address with unique port numbers The cluster address is used in entity and stateless EJBs to construct the host name portion of URLs. If the cluster address is not set, EJB handles may not work properly.

Table 6: Exemplary Cluster Configuration Parameters in an Embodiment

[0041]   In one embodiment, a Cluster Wizard/user interface user interface can prompt a user to assign the available Managed Server(s) to a cluster within the domain. This wizard/user interface is displayed only if a user have defined at least one cluster. The cluster wizard/user interface prompts a user to create an HTTP proxy application to proxy client requests to the cluster. An HTTP proxy application operates as an intermediary for HTTP requests.

[0042]   In one embodiment, another wizard/user interface prompts a user to define the configuration information for the Windows and UNIX machines in the domain (see Table 7). A user may want to perform this in circumstances such as (but not limited to) the following: an Administration Server uses the machine definition, in conjunction with the Node Manager application, to start remote servers; and a Web/app server uses configured machine names when determining which server in a cluster is best able to handle certain tasks, such as HTTP session replication. Web/app server then delegates those tasks to the identified server.

PARAMETER	DESCRIPTION
Name	Enter a valid machine name: a string of characters that can include spaces. The machine name:

PARAMETER	DESCRIPTION
	• Is used to identify the machine within the Web/app server domain; it is not required to match the network name for the machine. • Can be unique among all configuration component names within the domain.
Node Manager Listen Address (Optional)	Select a value from the drop-down list for the listen address used by the Node Manager. By default, the IP addresses defined for the local system and localhost are shown in the drop-down list. The default is localhost. Valid values for the listen address are as follows: • IP address of the computer that hosts the server • DNS name that resolves to the host • localhost • All Local Addresses
Node Manager Listen Port (Optional)	Enter a valid value for the listen port used by the Node Manager. Any number between 1 and 65535 is a valid value. The default value is 5555.

Table 7: Exemplary Machine Configuration Parameters in an Embodiment

[0043] In one embodiment, a user can configure a database service such as but not limited to JDBC. JDBC enables Java programmers to interact with common database management systems such as Oracle, Microsoft SQL Server, Sybase. A wizard/user interface gives the user the option of configuring a database service by defining several JDBC components in **Table 8**.

TASK	DESCRIPTION
Configuring JDBC Connection Pools (Optional)	Create ready-to-use pools of connections to a user DBMS. This task is a prerequisite for configuring JDBC MultiPools.
Configuring JDBC MultiPools (Optional)	Set up a group of JDBC connection pools to accommodate either a high-availability algorithm or a load-balancing algorithm.
Assigning JDBC Connection Pools to MultiPools (Optional)	Assign the JDBC connection pools to the defined JDBC MultiPools. This window only is displayed only if a user define one or more JDBC connection pools and JDBC MultiPools.
Configuring JDBC Data Sources	Configure the data sources that are bound to the JNDI tree and assign a JDBC connection pool. A data source object enables

TASK	DESCRIPTION
(Optional)	JDBC applications to obtain a DBMS connection from a connection pool.
Testing JDBC Connection Pools and Setting Up a JDBC Database(Optional)	Test a user JDBC connection pool configurations on the local machine.
Setting Up the JDBC Database Content (Optional)	Set up the database content used by the applications in a user domain using pre-existing SQL or database loading files. A configuration template may contain a set of SQL files organized by database type.

Table 8: Exemplary Tasks for Configuring JDBC in an Embodiment

[0044] In one embodiment, a user can configure JMS. A wizard/user interface gives a user access to enterprise messaging systems that make it possible for applications to communicate with one another. In one embodiment, JMS configuration parameters are listed in **Table 9**.

TASK	DESCRIPTION
Configuring JMS Connection Factories	Configure connection factories, JMS objects that encapsulate connection configuration information, and enable JMS applications to create connections.
Configuring JMS Destination Keys	Configure JMS destination keys that are used to define the sort order for messages that arrive at a specific destination.
Configuring JMS Templates	Configure JMS templates which provide an efficient means of defining multiple destinations (queues and topics) with similar configuration settings.
Assigning JMS Destination Keys	Assign destination keys to templates.
Configuring JMS File Stores	Configure JMS file stores. A JMS file store is a disk-based file that is used to store persistent messages.
Configuring JMS JDBC File Stores	Configure JMS JDBC stores, a JDBC-accessible database used to store persistent messages.
Configuring JMS Servers	Configure JMS servers to manage connections and message requests on behalf of clients.
Assigning JMS Servers to Web/app	Assign the JMS servers to Web/app server instances.

TASK	DESCRIPTION
server Instances	
Configuring JMS Topics	Configure JMS topics. JMS topics support the publish/subscribe (Pub/sub) messaging model which enables an application to send a message to multiple applications. Pub/sub messaging applications send and receive messages by subscribing to a topic.
Configuring JMS Queues	Configure JMS queues. JMS queues support the point-to-point (PTP) messaging model which enables one application to send a message to another application. PTP messaging applications send and receive messages using named queues.
Configuring JMS Distributed Topics	Configure JMS distributed topics, a set of physical topics that can support service continuity in the event of a Web/app server failure within a cluster.
Configuring JMS Distributed Queues	Configure JMS distributed queues, a set of physical queues that can support service continuity in the event of a Web/app server failure within a cluster.
Assigning JMS Distributed Destinations	Assign JMS distributed queues or topics to a Web/app server or cluster.
Configuring JMS Distributed Topic Members	Assign members to a particular JMS distributed topic.
Configuring JMS Distributed Queue Members	Assign members to the JMS distributed queue.

Table 9: Exemplary Tasks for Configuring JMS in an Embodiment

[0045]   Targeting of applications and services (e.g., JMS and JDBC) to servers or clusters is optional. A wizard/user interface (not shown) can allow a user to target the servers and clusters onto which a user want to deploy applications and services.

[0046]   In some situations, a user may need to extend an existing domain with an application, a component product, or a set of services. By way of a non-limiting example, if a user needs to develop an IDE application for a domain in which a user are already running a Web/app server application, a user can extend the functionality of the domain by adding IDE to it. The Configuration Wizard/user interface simplifies the task of extending an existing domain by using extension templates. An extension template can define applications and services that can be added to an existing domain. BEA delivers a set of predefined extension templates, which are described in Template Reference. **Table**

10 summarizes a procedure for extending an existing domain using the Configuration Wizard/user interface in one embodiment.

TASK	DESCRIPTION
Creating or Extending a Configuration	Choose whether to create a new WebLogic domain configuration or add to an existing domain configuration. To extend an existing domain configuration, choose Extend an existing WebLogic configuration.
Choosing a WebLogic Configuration Directory	Select the host directory for the domain a user want to update.
Selecting a Configuration Extension Template	Specify an extension (or application) template that allows a user to add applications and services to an existing domain.
Configuring JDBC When Extending a Domain	Define parameters for Java Database Connectivity (JDBC).
Configuring JMS When Extending a Domain	Define parameters for Java Message Service (JMS).
Targeting Applications and Services to Servers and Clusters When Extending a Domain	Define parameters for the target servers and clusters onto which a user want to deploy application components, such as Web applications and EJB modules, and application services, such as JDBC or JMS components, or startup and shutdown classes.
Configuring JMS Domain Creating A user WebLogic	Review a user current configuration settings and launch the process that updates the domain.

Table 10: Exemplary Tasks for Extending a Domain in an Embodiment

[0047]   A configuration template can define the full set of resources within a domain, including infrastructure components, applications, services, security options, and general environment and operating system parameters. In various embodiments, a Configuration Template Builder makes it easy to create templates. By way of a non-limiting example, the template builder enables:

- Definition and propagation of a standard domain across a development project
- Distribution of a domain along with an application that has been developed to run on that domain

[0048]   Any template a user create with the Configuration Template Builder can

be used as input to the Configuration Wizard/user interface. The Configuration Wizard/user interface uses it, in turn, as the basis for creating a domain that is customized for a user target environment. **Table 11** summarizes the procedure for creating a configuration template using the Configuration Template Builder.

STEP	DESCRIPTION
Creating a New Template	Choose the type of template a user want to create: configuration or extension. To create a configuration template, choose Create a Configuration Template.
Selecting a Template Configuration Source	Select the configuration template or the directory of the domain from which a user want to create a new configuration template.
Describing the Template	Specify a description of the template that will be displayed within the Select a Configuration Template window of the Configuration Wizard/user interface.
Adding Applications to A user Configuration Template	Review and modify, if desired, the list of applications to be included in the template.
Adding Files to A user Configuration Template	Review and modify, if desired, the files to be included in the template.
Adding SQL Scripts Into A user Configuration Template	Add SQL scripts for each database that a user expect to be used with the domains created from this template and specify the order in which the scripts are executed.
Configuring Managed Servers, Clusters, and Machines When Creating Configuration Templates	Optionally, define parameters for the Managed Servers, clusters, and host machines in a user domain.
Configuring JDBC When Creating Configuration Templates	Optionally, define parameters for Java Database Connectivity (JDBC).
Configuring JMS When Creating Configuration Templates	Optionally, define parameters for Java Message Service (JMS).
Targeting Servers and Clusters When Creating Configuration Templates	Optionally, define parameters for the target servers and clusters onto which a user want to deploy application components (such as Web applications and EJB modules) and application services (such as JDBC or JMS components), or startup and shutdown classes.
Configuring Security	Specify a user name and password to be used for starting the

STEP	DESCRIPTION
When Creating Configuration Templates	Administration Server and, if desired, configure additional security.
Building Start Menu Entries	Optionally, define entries for the Windows Start Menu.
Preparing Scripts and Files With Replacement Variables	Replace specific paths, filenames, and other configuration environment settings with replacement variables in text files contained in the template. The Configuration Wizard/user interface later substitutes the variables with exact strings to set up a specific WebLogic configuration.
Creating A user Configuration Template	Review the configuration template details, specify the name and directory for the configuration template, and launch the process that creates it.

Table 11: Tasks for Creating a Template in an Embodiment

[0049] In one embodiment, an extension template can define applications and services that can be used to extend an existing domain. This type of template can be used when a user is updating a domain. The applications and services stored in the selected extension templates can be imported into the domain using the Configuration Wizard/user interface. The Configuration Template Builder provides a simple means for creating extension templates which can be used as input to the Configuration Wizard/user interface. The process used to create an extension template is virtually the same as the process used to create a configuration template except that a user are not prompted to define any infrastructure components.

[0050] One embodiment includes a computer program product which is a storage medium (media) having instructions stored thereon/in which can be used to program a computer to perform any of the features presented herein. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

[0051] Stored on any one of the computer readable medium (media), the present invention includes software for controlling both the hardware of the general purpose/specialized computer or microprocessor, and for enabling the computer or

microprocessor to interact with a human user or other mechanism utilizing the results of the present invention. Such software may include, but is not limited to, device drivers, operating systems, execution environments/containers, and applications.

[0052]   The foregoing description of the preferred embodiments of the present invention have been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Many modifications and variations will be apparent to the practitioner skilled in the art. Embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention, the various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.